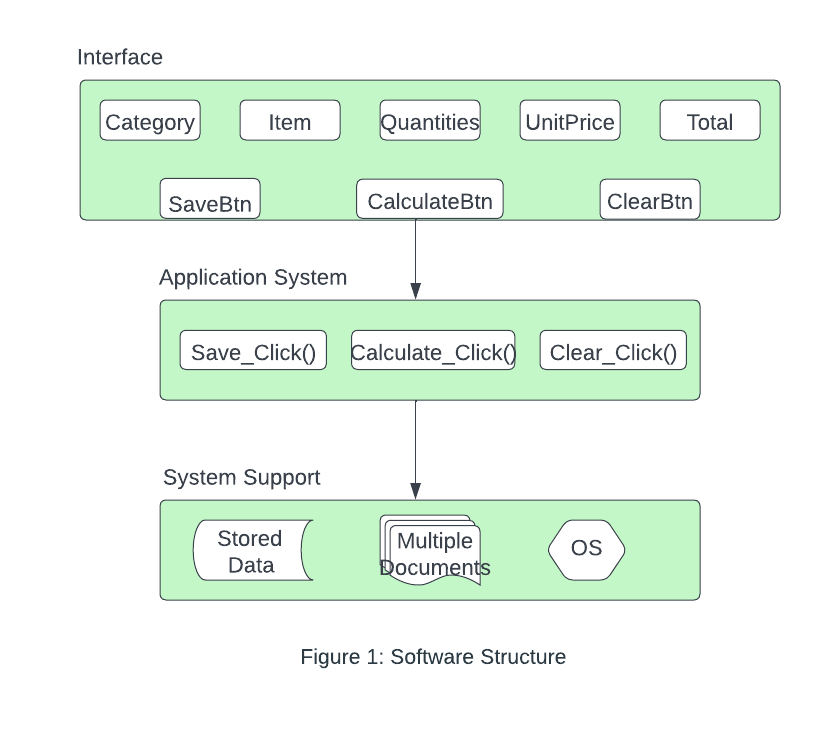
**Design Document**

**Introduction**

This project will be a windows form application run on Visual Studio in C#. The project will be a Material List Estimator. It will allow the user to pick Items from a list and the quantity of that said item. There will also be a budget per category option letting the user set their own budget per category. Once the user is done with the list it will calculate the total amount spent and display a pie chart depicting all items bought and the most expensive items.

**Architecture**

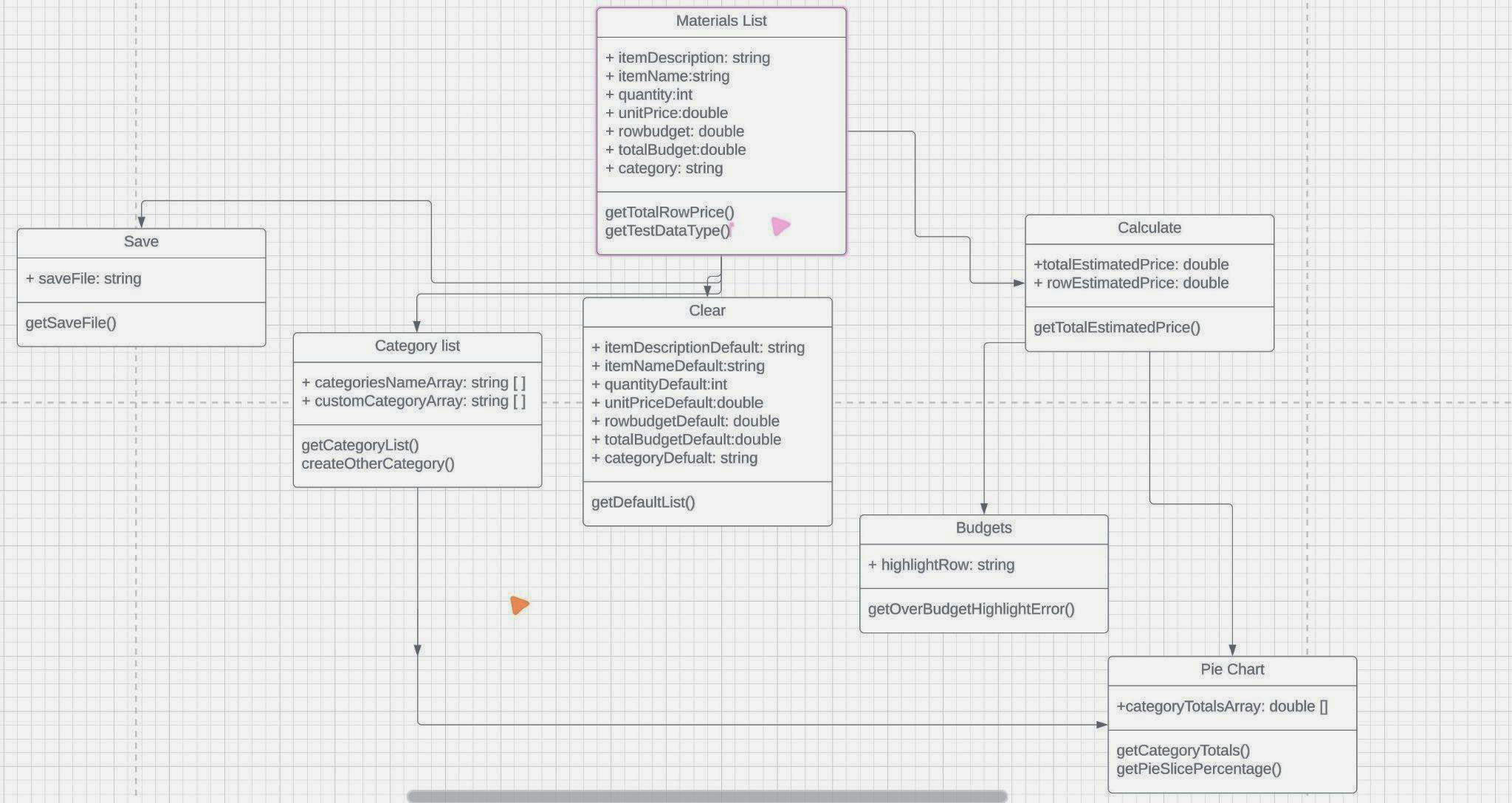
Our software architecture will be a Layered Architecture pattern. The structure is shown in Figure 1 below.



**Use Cases**

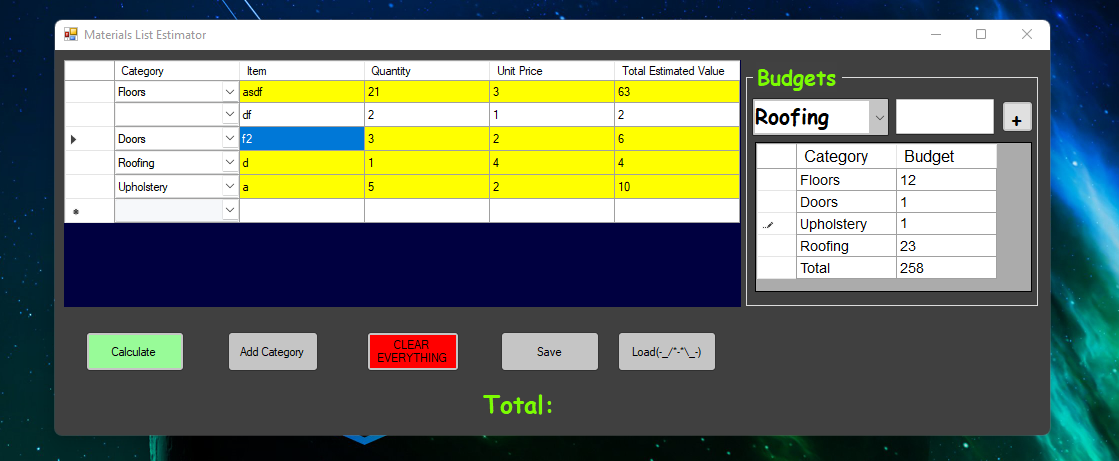
Diagram

Description automatically generated

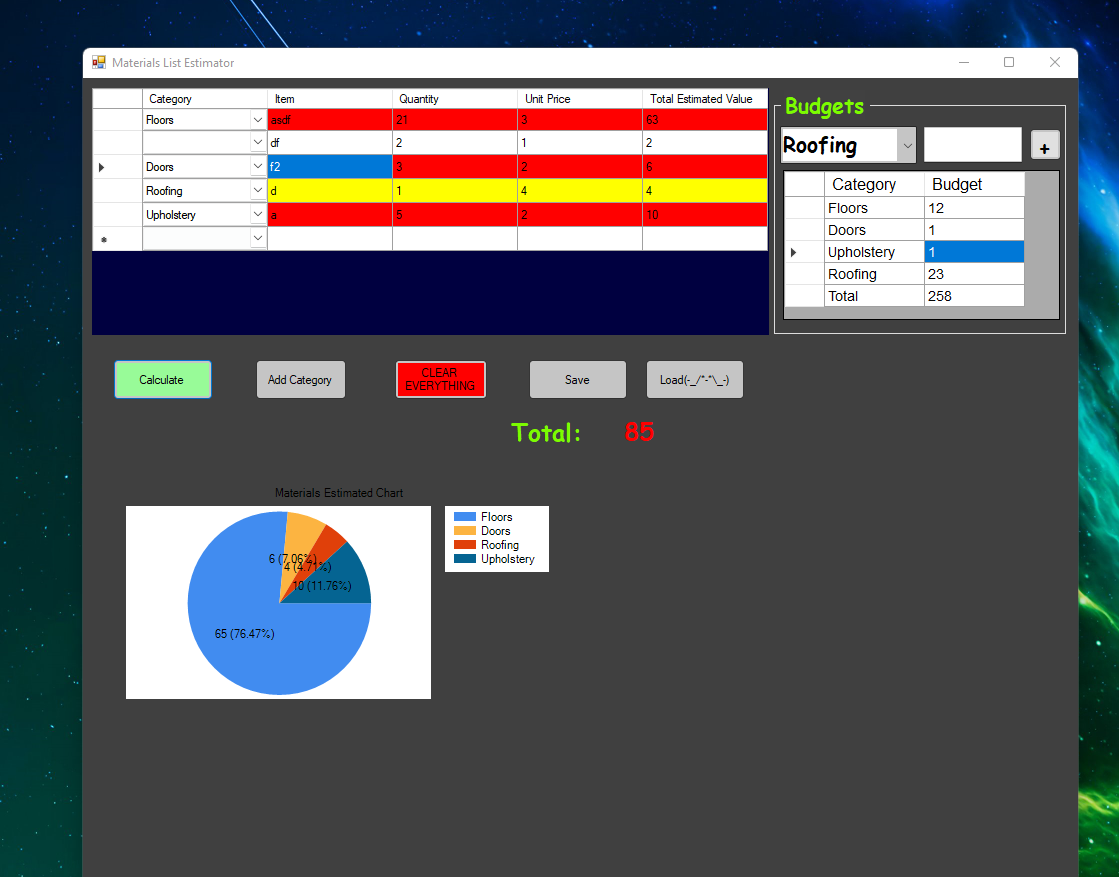
**UML**

**GUI SCREENSHOTS**

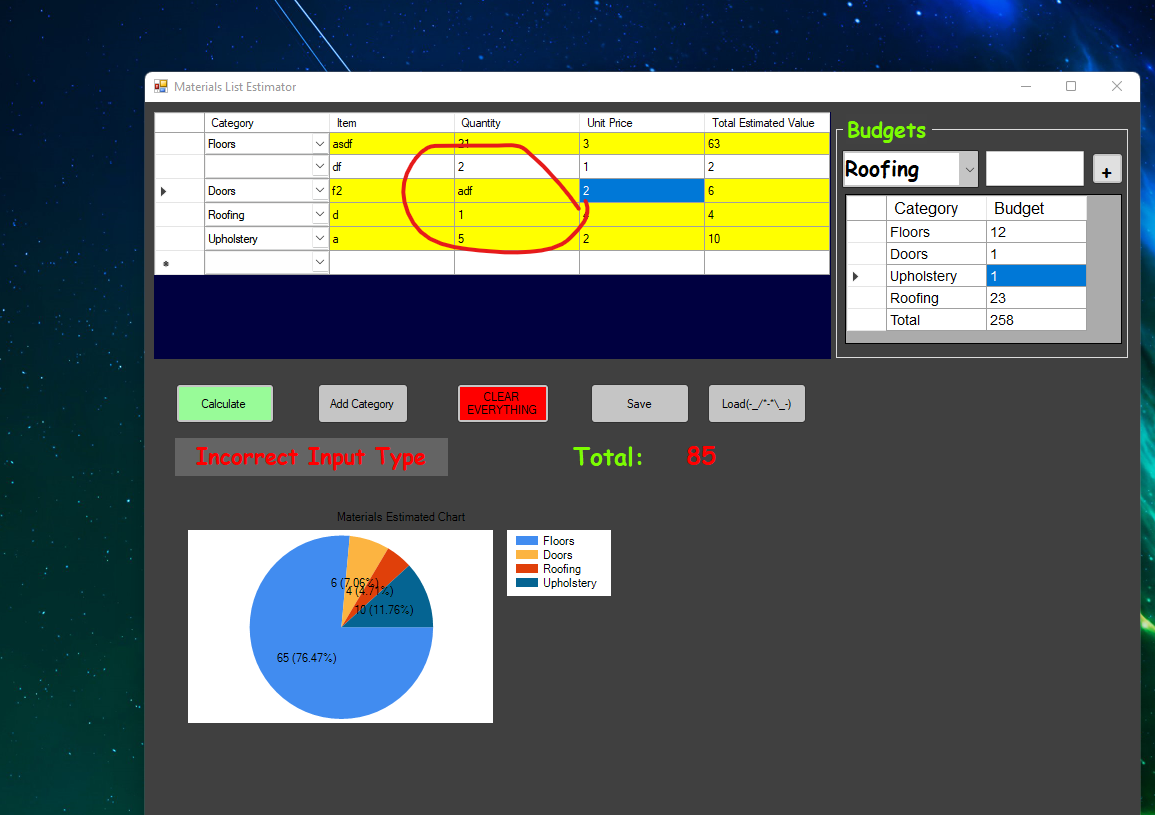
**ENTERING DATA**

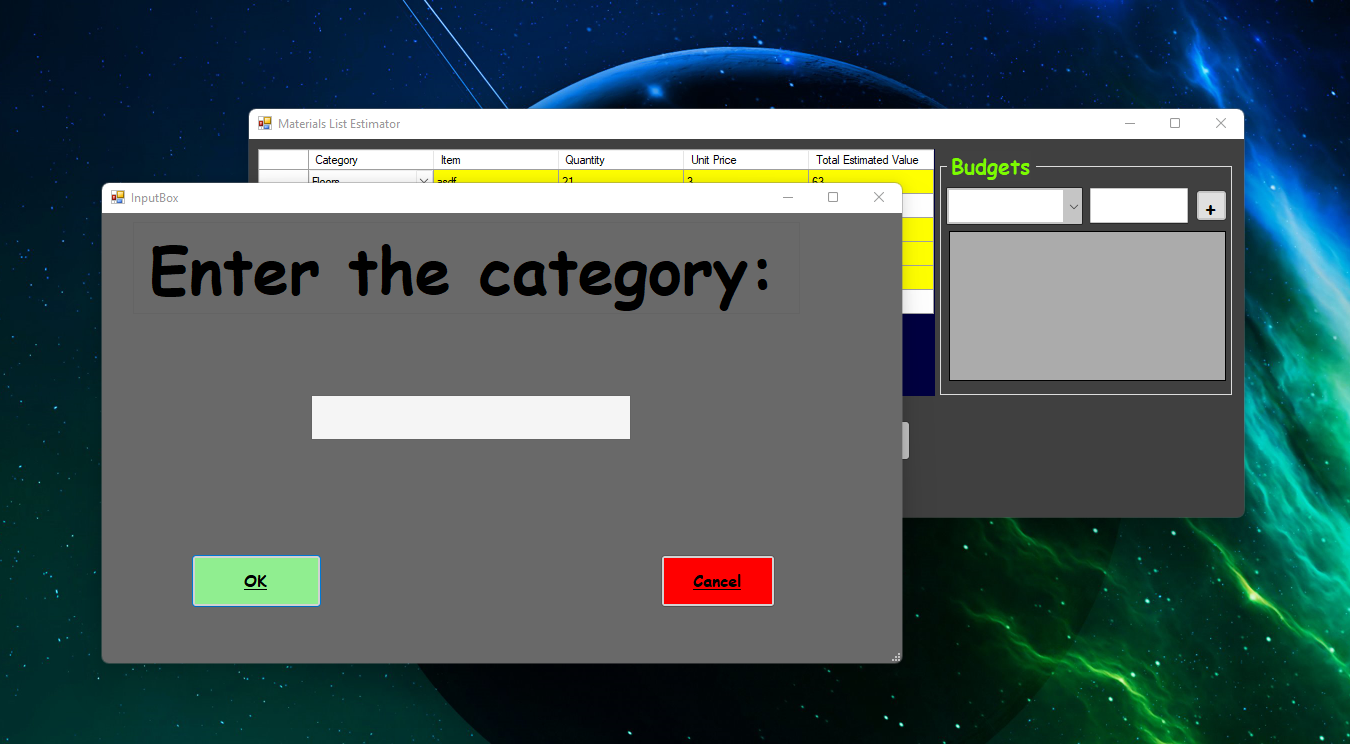
****

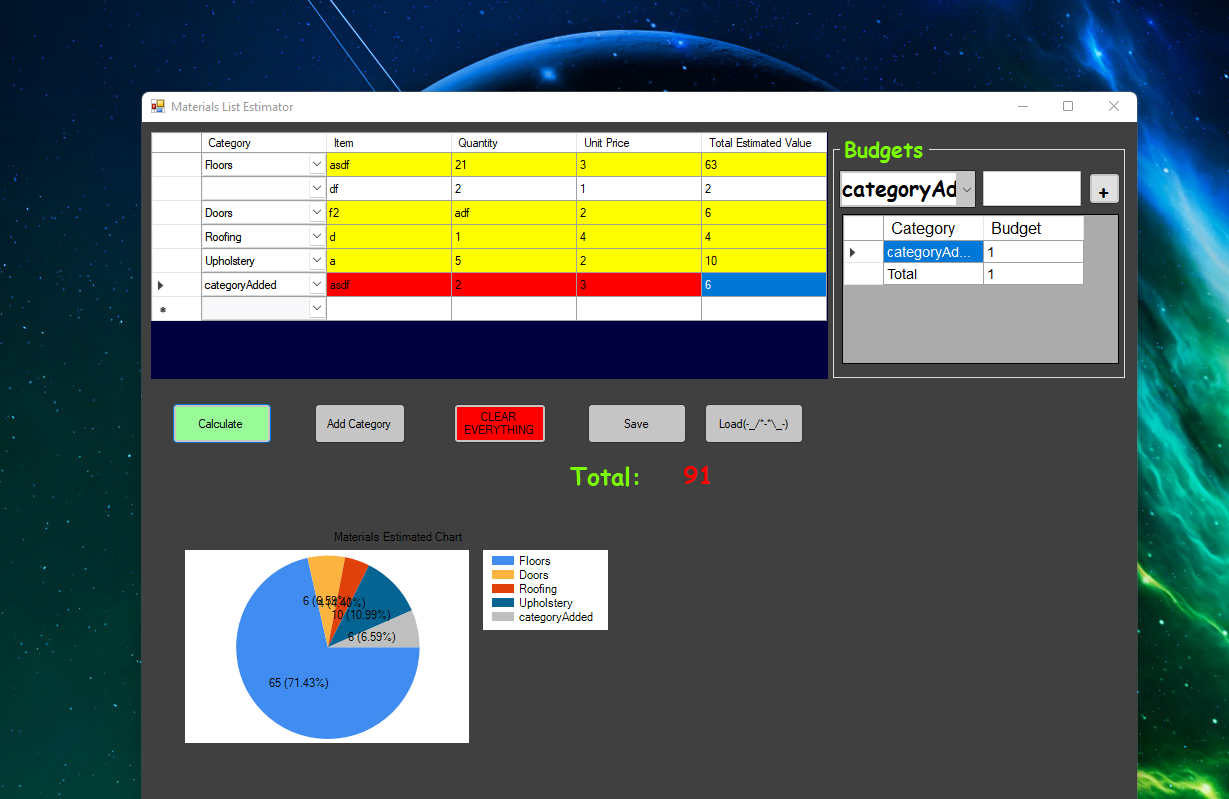
**Hitting Calcualte Button:**

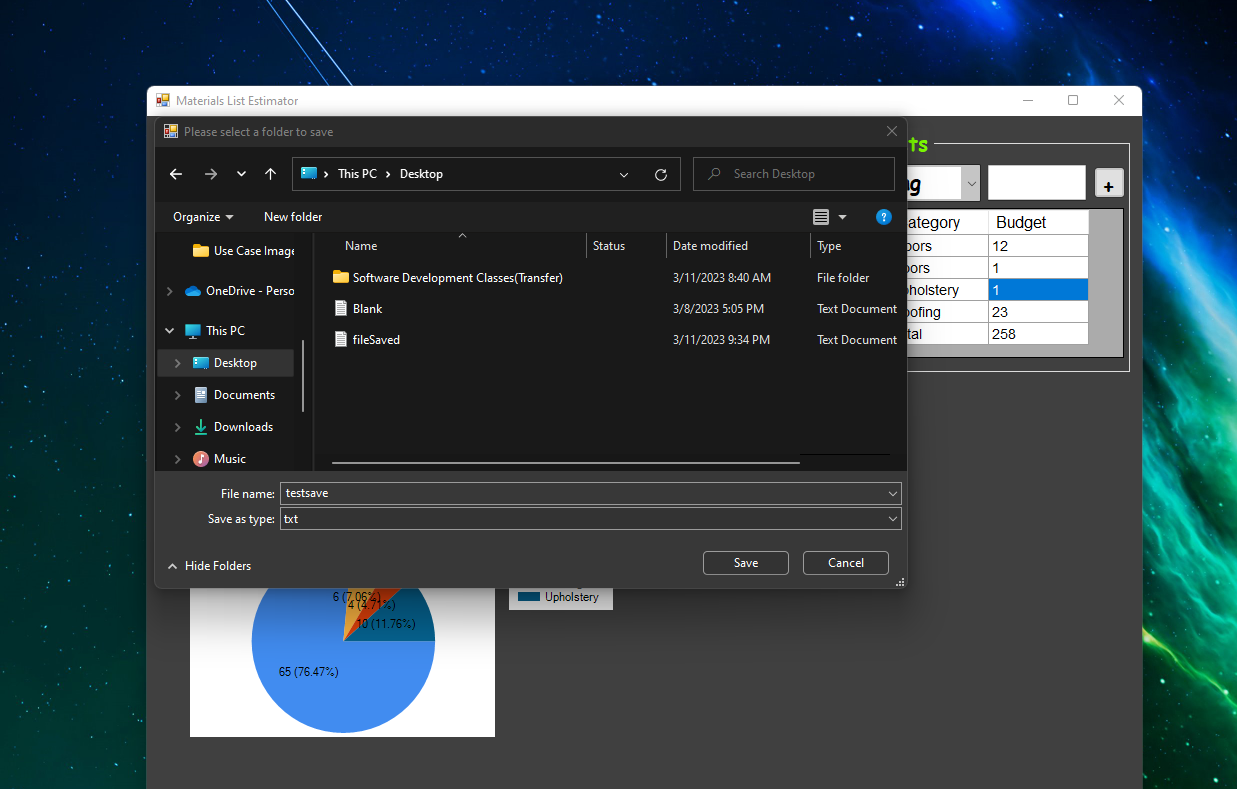


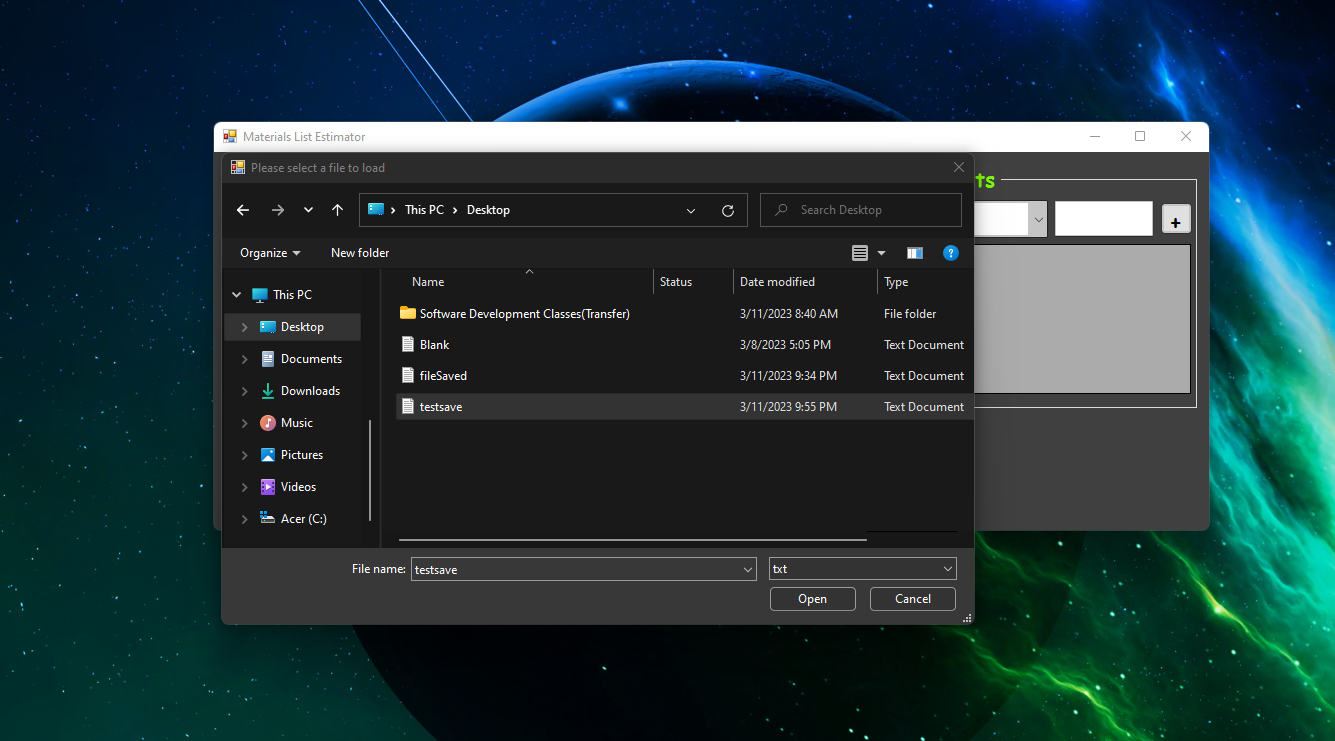
**WRONG INPUT TYPE:**

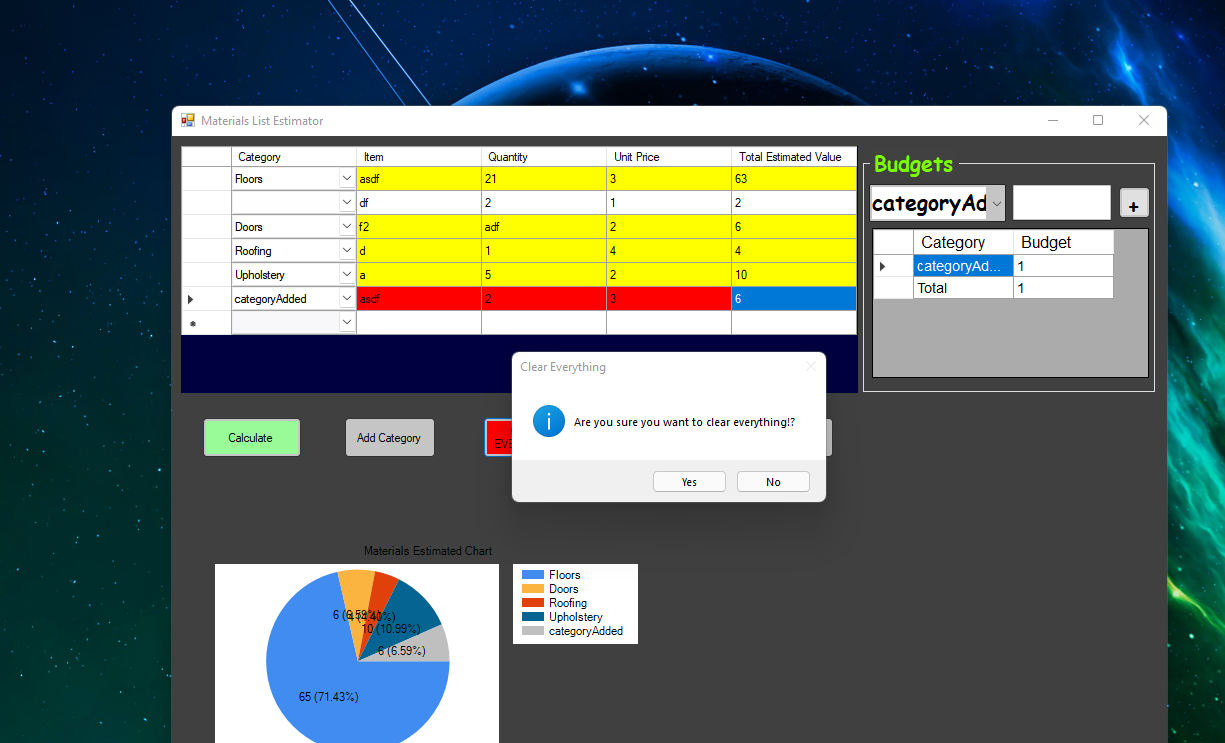
****

**ADDING NEW CATEGORY**

**CATEGORY ADDED**

**SAVING FILE**

**LOADING FILE**

**CLEARING EVERYTHING**

**Test Cases**

| **Case#** | **Case Description** | **Test Target** |
| --- | --- | --- |
| 1 | Adds up all the totals for each row and displays the correct total at the bottom. | To test if this section is working one of the group members can test multiple times per row and total. |
| 2 | Calculates the order quantity times the unit price and displays the correct total at the far right of each row. | Like the first activity a member of this group can manually calculate the totals multiple times to ensure the math is correct. |
| 3 | The clear button will clear the entire table and begin a new materials list. | Using the button will be the best way to test if it works, making sure the spreadsheet is fully cleared. |
| 4 | Save button will save the information to a file. | To test the save button we will type into the spreadsheet and save it. Then closing the program and re-opening the program to make sure the saved data is still there. |
| 5 | Calculate button will display the total estimated cost, budget exceeded, and pie chart. | To test the calculate button we can click the button and see the output and make sure it matched up with our own physical calculations. This test will be done countless times to ensure the correct totals every time. |
| 6 | Text boxes will allow the user to enter type letters and numbers into the item description and item. However, it will only allow the user to enter numbers into quantity, unit price, and budget. If the user tries to type letters into the text boxes, it will inform the user that letters are not allowed via a prompt. | The text boxes will be checked when running the program to make sure they are working properly. To check the quantity error code when typing a letter instead of a number we will just type into the text box to make sure it is working. |
| 7 | The drop-down list will be enabled on the far left of the grid and will allow the user to select from a variety of categories successfully. | To test this function, we will manually click the drop-down list and make sure it successfully selects a category. |
| 8 | Our entire application will be built in a grid in a similar manner to how excel grid looks. | We will make sure the spreadsheet looks and functions like Excel. |
| 9 | The pie chart will use the total of each category and divide it by the total estimated cost determining the percent sliver of each slice on the pie graph. | When displaying the pie chart, we will manually calculate the totals per percentage to make sure they match with the pie chart. We will also test to make sure each slice of the pie represents the percentage correctly. |
| 10 | Allow the user to input a budget for each category and then a budget for overall and if the user exceeds the budget when they hit calculate, it will highlight the category that exceeds the budget in the red. | We will make a mock spreadsheet with budgets per category and test if the budget system does turn red when over. |
| 11 | When the user hits the calculate button, it will highlight any category that exceeds their budget or which category that is most expensive in red. | We will make a mock spreadsheet with budgets per category and test if the budget system does turn when the calculate button is pressed. |
| 12 | There will be a textbox that will allow the user to add a new category to the drop list if they don’t like the current categories given. | This will be tested by clicking on the textbox and seeing if a category and items appear in the correct spot and are fully functional. |

Note: Each case matches each requirement.

**Summary**

This project will be a Material List Estimator, it will be designed and coded in Visual studio in C#. This project will allow a user to enter items into a spreadsheet and pick the quantities of the said items. Once the user is finished with the spreadsheet it can calculate the total cost of items and display a chart showing the most expensive to least expensive items that were picked by the user.